(http://twitter.com/intent/tweet?status=ORACLE-BASE%20-%20FTP%20From%20PL%2FSQL+https%3A%2F%2Foracle-base.com%2Farticles%2Fmisc%2Fftp-from-plsql) (http://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Foracle-base.com%2Farticles%2Fmisc%2Fftp-from-plsql&title=ORACLE-BASE%20-%20FTP%20From%20PL%2FSQL) (https://plus.google.com/share?url=https%3A%2F%2Foracle-base.com%2Farticles%2Fmisc%2Fftp-from-plsql) (http://www.linkedin.com/shareArticle?mini=true&url=https%3A%2F%2Foracle-base.com%2Farticles%2Fmisc%2Fftp-from-plsql&title=ORACLE-BASE%20-%20FTP%20From%20PL%2FSQL&source=oracle-base.com) (http://www.reddit.com/submit?url=https%3A%2F%2Foracle-base.com%2Farticles%2Fmisc%2Fftp-from-plsql&title=ORACLE-BASE%20-%20FTP%20From%20PL%2FSQL)

Home (/) » Articles (/articles) » Misc (/articles/misc) » Here

FTP From PL/SQL

Sometimes it's preferable to trigger FTP jobs directly from PL/SQL rather than rely on CRON or AT. This article contains a brief description of the two methods I use.

- Shell Script
- PL/SQL FTP API
- ACL for 11g
- SFTP and FTPS

Shell Script

The first method relies on a java stored procedure, described in Shell Commands From PL/SQL (/articles/8i/shell-commands-from-plsql), which can be used to trigger a shell script to perform the transfer. The shell script may look like the following.

```
#! /bin/ksh

# Move to appropriate directory on local server
cd /extracts

# FTP all files in directory
ftp -inv ftp.company.com <<EOF
user ftpuser ftppassword
# Move to appropriate directory on remote server.
cd /loads
ascii
mput *.*
bye
EOF
```

PL/SQL FTP API

The second approach uses a combination of the UTL_TCP (http://docs.oracle.com/cd/E11882_01/appdev.112/e40758/u_tcp.htm) and UTL_FILE (http://docs.oracle.com/cd/E11882_01/appdev.112/e40758/u_file.htm) packages to create a simple FTP API (ftp.pks (/dba/miscellaneous/ftp.pks), ftp.pkb (/dba/miscellaneous/ftp.pkb)). Once the API is loaded into the appropriate schema simple FTP commands can be initiated as follows.

```
CREATE OR REPLACE DIRECTORY my_docs AS '/u01/app/oracle/';
SET SERVEROUTPUT ON SIZE 1000000
@c:\ftp.pks
@c:\ftp.pkb

-- Retrieve an ASCII file from a remote FTP server.
DECLARE
  l_conn  UTL_TCP.connection;
BEGIN
  l_conn := ftp.login('ftp.company.com', '21', 'ftpuser', 'ftppassword');
  ftp.ascii(p_conn => l_conn);
  ftp.get(p_conn      => l_conn,
          p_from_file => '/u01/app/oracle/test.txt',
          p_to_dir    => 'MY_DOCS',
          p_to_file   => 'test_get.txt');
  ftp.logout(l_conn);
END;
/

-- Send an ASCII file to a remote FTP server.
DECLARE
  l_conn  UTL_TCP.connection;
BEGIN
  l_conn := ftp.login('ftp.company.com', '21', 'ftpuser', 'ftppassword');
  ftp.ascii(p_conn => l_conn);
  ftp.put(p_conn      => l_conn,
          p_from_dir  => 'MY_DOCS',
          p_from_file => 'test_get.txt',
          p_to_file   => '/u01/app/oracle/test_put.txt');
  ftp.logout(l_conn);
END;
/

-- Retrieve a binary file from a remote FTP server.
DECLARE
  l_conn  UTL_TCP.connection;
BEGIN
  l_conn := ftp.login('ftp.company.com', '21', 'ftpuser', 'ftppassword');
  ftp.binary(p_conn => l_conn);
  ftp.get(p_conn      => l_conn,
          p_from_file => '/u01/app/oracle/product/9.2.0.1.0/sysman/reporting/gif/jobs
          p_to_dir    => 'MY_DOCS',
          p_to_file   => 'jobs_get.gif');
  ftp.logout(l_conn);
END;
/

-- Send a binary file to a remote FTP server.
DECLARE
  l_conn  UTL_TCP.connection;
```

```
BEGIN
  l_conn := ftp.login('ftp.company.com', '21', 'ftpuser', 'ftppassword');
  ftp.binary(p_conn => l_conn);
  ftp.put(p_conn      => l_conn,
          p_from_dir  => 'MY_DOCS',
          p_from_file => 'jobs_get.gif',
          p_to_file   => '/u01/app/oracle/jobs_put.gif');
  ftp.logout(l_conn);
END;
/

-- Get a directory listing from a remote FTP server.
DECLARE
  l_conn  UTL_TCP.connection;
  l_list  ftp.t_string_table;
BEGIN
  l_conn := ftp.login('ftp.company.com', '21', 'ftpuser', 'ftppassword');
  ftp.list(p_conn   => l_conn,
           p_dir   => '/u01/app/oracle',
           p_list  => l_list);
  ftp.logout(l_conn);

  IF l_list.COUNT > 0 THEN
    FOR i IN l_list.first .. l_list.last LOOP
      DBMS_OUTPUT.put_line(i || ': ' || l_list(i));
    END LOOP;
  END IF;
END;
/

-- Get a directory listing (file names only) from a remote FTP server.
DECLARE
  l_conn  UTL_TCP.connection;
  l_list  ftp.t_string_table;
BEGIN
  l_conn := ftp.login('ftp.company.com', '21', 'ftpuser', 'ftppassword');
  ftp.nlst(p_conn   => l_conn,
           p_dir   => '/u01/app/oracle',
           p_list  => l_list);
  ftp.logout(l_conn);

  IF l_list.COUNT > 0 THEN
    FOR i IN l_list.first .. l_list.last LOOP
      DBMS_OUTPUT.put_line(i || ': ' || l_list(i));
    END LOOP;
  END IF;
END;
/

-- Rename a file on a remote FTP server.
DECLARE
```

```
  l_conn  UTL_TCP.connection;
BEGIN
  l_conn := ftp.login('ftp.company.com', '21', 'ftpuser', 'ftppassword');
  ftp.rename(p_conn => l_conn,
             p_from => '/u01/app/oracle/dba/shutdown',
             p_to   => '/u01/app/oracle/dba/shutdown.old');
  ftp.logout(l_conn);
END;
/

-- Delete a file on a remote FTP server.
DECLARE
  l_conn  UTL_TCP.connection;
BEGIN
  l_conn := ftp.login('ftp.company.com', '21', 'ftpuser', 'ftppassword');
  ftp.delete(p_conn => l_conn,
             p_file => '/u01/app/oracle/dba/temp.txt');
  ftp.logout(l_conn);
END;
/

-- Create a directory on a remote FTP server.
DECLARE
  l_conn  UTL_TCP.connection;
BEGIN
  l_conn := ftp.login('ftp.company.com', '21', 'ftpuser', 'ftppassword');
  ftp.mkdir(p_conn => l_conn,
            p_dir => '/u01/app/oracle/test');
  ftp.logout(l_conn);
END;
/

-- Remove a directory from a remote FTP server.
DECLARE
  l_conn  UTL_TCP.connection;
BEGIN
  l_conn := ftp.login('ftp.company.com', '21', 'ftpuser', 'ftppassword');
  ftp.rmdir(p_conn => l_conn,
            p_dir  => '/u01/app/oracle/test');
  ftp.logout(l_conn);
END;
/
```

The basic functions are implemented using LOBs to allow FTP without having to access files on the local filesystem. The `get` and `put` procedures string these together to form a complete job using all the functions. If a straight forward FTP to, or from, the local filesystem is required it is more efficient to use the `GET_DIRECT` and `PUT_DIRECT` procedures as they avoid the temporary LOBs.

The current implementation has the following issues:

- The mput and mget operations are not supported directly, but can be implemented using a combination of the list/nlst and get/put operations.

- The implementation of binary transfers relies on `UTL_FILE` features only available in Oracle9i Release 2 upwards.
- There is no support for ASCII mode in the `PUT_DIRECT` procedure.

Thanks to Hans van Doormalen for noticing I wasn't closing my passive connections. I do now :)

ACL for 11g

The introduction of Fine-Grained Access to Network Services in Oracle Database 11g Release 1 (/articles/11g/fine-grained-access-to-network-services-11gr1) means you will need to configure an access control list (ACL) to allow UTL_TCP to access the network. The examples above work correctly with the following basic ACL. You will need to amend the FTP server details and username details to match your FTP server address and the Oracle username running the FTP API.

```
DECLARE
  l_acl_name         VARCHAR2(30) := 'utl_tcp.xml';
  l_ftp_server_ip    VARCHAR2(20) := '192.168.0.131';
  l_ftp_server_name  VARCHAR2(20) := 'ftp.company.com';
  l_username         VARCHAR2(30) := 'TEST';
BEGIN
  DBMS_NETWORK_ACL_ADMIN.create_acl (
    acl          => l_acl_name,
    description  => 'Allow connections using UTL_TCP',
    principal    => l_username,
    is_grant     => TRUE,
    privilege    => 'connect',
    start_date   => SYSTIMESTAMP,
    end_date     => NULL);

  COMMIT;

  DBMS_NETWORK_ACL_ADMIN.add_privilege (
    acl         => l_acl_name,
    principal   => l_username,
    is_grant    => FALSE,
    privilege   => 'connect',
    position    => NULL,
    start_date  => NULL,
    end_date    => NULL);

  COMMIT;

  DBMS_NETWORK_ACL_ADMIN.assign_acl (
    acl         => l_acl_name,
    host        => l_ftp_server_ip,
    lower_port  => NULL,
    upper_port  => NULL);

  DBMS_NETWORK_ACL_ADMIN.assign_acl (
    acl         => l_acl_name,
    host        => l_ftp_server_name,
    lower_port  => NULL,
    upper_port  => NULL);

  COMMIT;
END;
/
```

SFTP and FTPS

The world has changed drastically since this article was first written. If you are using FTP today, you are probably making a very big mistake. This leaves you with two FTP-like alternatives.

- FTPS : This is FTP, but using a certificate to encrypt the communication. It requires a suitable FTPS service to be present on the server, so it is not very popular. Anton Scheffer (https://technology.amis.nl/author/anton-scheffer/) wrote about accessing FTPS directly from PL/SQL (https://technology.amis.nl/2014/10/01/ftps-plsql/). It is essentially what I've done for FTP, with the certificate and wallet thrown in the mix.

- SFTP : This is basically mimicking FTP through SSH. It feels like FTP, but it is actually just a connection to the SSH service on the server, so no additional software is needed. If you need an FTP-like interface, this is probably the best route to go. I've not seen a SFTP implementation from PL/SQL, but I will explain below how I implement this.

To use SFTP from PL/SQL I do the following.

- Use keypair authentication between the database server and the remote SSH service, allowing passwordless connections (described here (/articles/linux/user-equivalence-configuration-on-linux)).
- Write a shell script to perform the SFTP command.
- Call the shell script from a Java Stored Procedure (/articles/8i/shell-commands-from-plsql), or preferably an executable job using the scheduler (/articles/10g/scheduler-10g).

For more information see:

- Shell Commands From PL/SQL (/articles/8i/shell-commands-from-plsql)
- UTL_FILE Enhancements (/articles/9i/utl_file-enhancements-9i)
- UTL_TCP (http://docs.oracle.com/cd/E11882_01/appdev.112/e40758/u_tcp.htm)
- UTL_FILE (http://docs.oracle.com/cd/E11882_01/appdev.112/e40758/u_file.htm)
- Fine-Grained Access to Network Services in Oracle Database 11g Release 1 (/articles/11g/fine-grained-access-to-network-services-11gr1)

Hope this helps. Regards Tim...

Back to the Top.

75 comments, read/add them... (/misc/comments?page_id=867)

(http://twitter.com/intent/tweet?status=ORACLE-BASE%20-%20FTP%20From%20PL%2FSQL+https%3A%2F%2Foracle-base.com%2Farticles%2Fmisc%2Fftp-from-plsql) (http://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Foracle-base.com%2Farticles%2Fmisc%2Fftp-from-plsql&title=ORACLE-BASE%20-%20FTP%20From%20PL%2FSQL) (https://plus.google.com/share?url=https%3A%2F%2Foracle-base.com%2Farticles%2Fmisc%2Fftp-from-plsql) (http://www.linkedin.com/shareArticle?mini=true&url=https%3A%2F%2Foracle-base.com%2Farticles%2Fmisc%2Fftp-from-plsql&title=ORACLE-BASE%20-%20FTP%20From%20PL%2FSQL&source=oracle-base.com) (http://www.reddit.com/submit?url=https%3A%2F%2Foracle-base.com%2Farticles%2Fmisc%2Fftp-from-plsql&title=ORACLE-BASE%20-%20FTP%20From%20PL%2FSQL)

Home (/) | Articles (/articles/articles) | Scripts (/dba/scripts) | Blog (/blog/) | Certification (/misc/ocp-

certification) | Misc (/misc/miscellaneous) | About (/misc/site-info)

About Tim Hall (/misc/site-info#biog)
Copyright & Disclaimer (/misc/site-info#copyright)